

03-29-00

A

UTILITY PATENT APPLICATION TRANSMITTAL
(Large Entity)*(Only for new nonprovisional applications under 37 CFR 1.53(b))*Docket No.
POU9-2000-0030-US1

Total Pages in this Submission

TO THE ASSISTANT COMMISSIONER FOR PATENTSBox Patent Application
Washington, D.C. 20231

Transmitted herewith for filing under 35 U.S.C. 111(a) and 37 C.F.R. 1.53(b) is a new utility patent application for an invention entitled:

STORAGE ISOLATION EMPLOYING SECURED SUBSPACE FACILITY

and invented by:

Carl E. Clark, Steven J. GreenspanIf a **CONTINUATION APPLICATION**, check appropriate box and supply the requisite information:☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No.: _____

Which is a:

☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No.: _____

Which is a:

☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No.: _____

Enclosed are:

Application Elements

1. ☒ Filing fee as calculated and transmitted as described below
2. ☒ Specification having 29 pages and including the following:
 - a. ☒ Descriptive Title of the Invention
 - b. ☐ Cross References to Related Applications *(if applicable)*
 - c. ☐ Statement Regarding Federally-sponsored Research/Development *(if applicable)*
 - d. ☐ Reference to Microfiche Appendix *(if applicable)*
 - e. ☒ Background of the Invention
 - f. ☒ Brief Summary of the Invention
 - g. ☒ Brief Description of the Drawings *(if drawings filed)*
 - h. ☒ Detailed Description
 - i. ☒ Claim(s) as Classified Below
 - j. ☒ Abstract of the Disclosure

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Total Pages in this Submission

3. ☒ Drawing(s) (when necessary as prescribed by 35 USC 113)

4. ☒ Oath or Declaration

- d. ☐ DELETION OF INVENTOR(S)
Signed statement attached deleting inventor(s) named in the prior application,
see 37 C.F.R. 1.63(d)(2) and 1.33(b).

6. ☐ Computer Program in Microfiche (*Appendix*)

7. ☐ Nucleotide and/or Amino Acid Sequence Submission (if applicable, all must be included)

- a. ☐ Paper Copy
- b. ☐ Computer Readable Copy (*identical to computer copy*)
- c. ☐ Statement Verifying Identical Paper and Computer Readable Copy

8. ☐ Assignment Papers (cover sheet & document(s))

9. ☐ 37 CFR 3.73(B) Statement (when there is an assignee)

10. * ☐ English Translation Document (if applicable)

11. ☒ Information Disclosure Statement/PTO-1449 ☒ Copies of IDS Citations

12. ☐ Preliminary Amendment

13. ☒ Acknowledgment postcard

14. ☒ Certificate of Mailing

- ☐ First Class ☒ Express Mail (Specify Label No.): EL497287185US

UTILITY PATENT APPLICATION TRANSMITTAL
(Large Entity)

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Docket No.
POU9-2000-0030-US1

Total Pages in this Submission

Accompanying Application Parts (Continued)

15. ☐ Certified Copy of Priority Document(s) *(if foreign priority is claimed)*

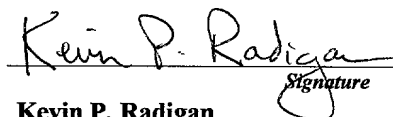
16. ☐ Additional Enclosures *(please identify below):*

Fee Calculation and Transmittal

CLAIMS AS FILED

For	#Filed	#Allowed	#Extra	Rate	Fee
Total Claims	31	- 20 =	11	x \$18.00	\$198.00
Indep. Claims	4	- 3 =	1	x \$78.00	\$78.00
Multiple Dependent Claims (check if applicable) <input type="checkbox"/>					\$0.00
BASIC FEE					\$690.00
OTHER FEE (specify purpose) _____					\$0.00
TOTAL FILING FEE					\$966.00

- ☐ A check in the amount of _____ to cover the filing fee is enclosed.
- ☒ The Commissioner is hereby authorized to charge and credit Deposit Account No. **09-0463 (IBM)** as described below. A duplicate copy of this sheet is enclosed.
- ☒ Charge the amount of **\$966.00** as filing fee.
 - ☒ Credit any overpayment.
 - ☒ Charge any additional filing fees required under 37 C.F.R. 1.16 and 1.17.
 - ☐ Charge the issue fee set in 37 C.F.R. 1.18 at the mailing of the Notice of Allowance, pursuant to 37 C.F.R. 1.311(b).


Signature

Kevin P. Radigan
Reg. No. 31,789
Attorney for Applicants
HESLIN & ROTHENBERG, P.C.
5 Columbia Circle
Albany, NY 12203
Telephone (518)452-5600
Facsimile (518)452-5579

Dated: **March 28**, 2000

cc:

CERTIFICATE OF MAILING BY "EXPRESS MAIL"

In Re Application of: Clark et al.

Title: STORAGE ISOLATION EMPLOYING SECURED SUBSPACE FACILITY

Attorney Docket No.: POU9-2000-0030-US1

"EXPRESS MAIL" MAILING LABEL NO. EL497287185US

Date of Deposit March 28, 2000

I hereby certify that this paper is being deposited with the U.S. Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and addressed to Box PATENT APPLICATION, Assistant Commissioner For Patents, Washington, D.C. 20231.

Robyn Dunlavey

(Typed or printed name of person mailing paper or fee)

Robyn Dunlavey

(Signature of person mailing paper or fee)

- Enclosures:
- * Utility Patent Application Transmittal (Large Entity) (3 pp.) (in duplicate)
 - * U.S. Patent Application which includes: (Specification - 18 pp.; 31 Claims - 10 pp.; Abstract - 1 p.).
 - * Declaration and Power of Attorney For Patent Application (3 pp.) (unsigned)
 - * Information Disclosure Citation (1 p.) w/references (2 cited)
 - * Nine (9) sheets of formal drawings
 - * Two (2) Postcards

**STORAGE ISOLATION EMPLOYING
SECURED SUBSPACE FACILITY**

Technical Field

5 This invention enhances the reliability of computer
system operation by isolating data (including programs) in
virtual subspaces from programs and other virtual subspaces
in the same subspace group. More particularly, this
invention ensures subspace isolation notwithstanding
10 execution of applications in address register addressing
mode.

Background of the Invention

15 It is common to have a family of programs and data that
are intertwined in their relationship and their execution,
such that a high rate of switching is essential among the
different programs and there is shared use of the databases
in the family. Such a family of programs and data are often
supported by a software subsystem (operating under an
operating system). The subsystem often handles a large
number of transactions that are concurrently accessing a
20 large number of different programs and databases in the
family. For example, the transactions of banking tellers
(both humans and machines) in a multi-branch banking complex
may concurrently use deposit programs and withdrawal
programs (that share the same database, i.e., customer
25 accounts), credit check programs and their databases, and
numerous other related banking programs and databases, all
of which are being accessed concurrently by a set of
transaction programs invoked by individual requests for
service.

Such programs and data have been found to be usable at their fastest potential rate when they are all in a single address space (AS) being accessed from one or more CPUs. However, subsequent experience has indicated significant failures in the execution of such programs, due to incorrect store operations by an executing program wiping out part of another or a database. Such execution failures have temporarily terminated the operation of a multi-branch banking business dependent on such a system. A programming system failure that causes a temporary outage of an entire business is usually considered a non-tolerable option, regardless of its speed of operation. Also, incorrect store operations that do not result in a system failure, but invalidly modify data and perpetuate without being detected are non-tolerable, difficult to detect program failures.

One way to prevent such program failures is to isolate the different programs and databases from each other in their system, so that one program cannot access another program or database in the system. One such storage isolation approach is presented in commonly assigned United States Letters Patent 5,361,356, by Clark et al., entitled "Storage Isolation With Subspace-Group Facility," the entirety of which is hereby incorporated herein by reference.

In this prior patent, a Branch in Subspace Group (BSG) instruction is executed in problem state (for example by an application program) for providing a fast instruction branch between address spaces within a restricted group of address spaces called a subspace group. The subspace group contains two types of address spaces: a base space and any number of subspaces. The subspace group is set up in a control table associated with each dispatchable unit (DU). This DU

control table contains: an identifier of a base space, an identifier of an access list that contains identifiers of all subspaces in the subspace group, an indicator of whether CPU control was last given to a subspace or to the base space, and an identifier of a last entered subspace in the group. The BSG instruction has an operand defining a general register containing the target virtual address and an associated access register containing an access-list-entry token (ALET) defining the target address space. The ALET indexes to a target subspace identifier in the access list, and then the associated virtual address locates the target instruction in the identified target address space. BSG instruction execution controls restrict the BSG branching only to an instruction in the subspace group.

Applicants have discovered that one restriction on the above-noted process is that secured subspace isolation was achieved only for primary and secondary space addressing, and not achieved for access register addressing while running with subspace active. Prohibiting the usage of access register addressing allows for a secure subspace environment, but limits the general applicability of secured subspaces on many systems, such as an International Business Machines S/390 Operating System. IBM markets a transaction manager which runs on S/390 and is referred to as Customer Information Systems (CICS). A CICS's direction to use subspaces for transaction isolation within an open transaction environment (OTE) for CICS applications would not be consistent nor acceptable unless access register addressing is available for the CICS applications and resource managers that the applications called.

In view of the above, applicants have discovered there is a need in the art to further extend the teachings of

subspace isolation to allow usage thereof in an access register addressing mode with secured subspace isolation.

Disclosure of the Invention

5 The shortcomings of the prior art are overcome and additional advantages are provided through the provision of a method for producing secured subspaces for transactions to be run. The method includes: from an operating system task, attaching a subtask that will restrict application
10 addressing; and wherein the attaching includes defining a subspace address environment as home space within a dispatchable unit access list associated with the subtask.

15 In another aspect, this invention provides at least one program storage device readable by a machine, tangibly embodying at least one program of instructions executable by the machine to perform a method for producing a secure subspace for a transaction. The method includes: from an
20 operating system task, attaching a subtask that will restrict application addressing; and wherein the attaching includes defining a subspace address environment as home space within a dispatchable unit access list associated with the subtask.

25 In a further aspect, a system is provided for producing a secure subspace for a transaction. The system includes means for attaching, from an operating system task, a subtask that will restrict application addressing. The means for attaching includes means for defining a subspace address environment within a dispatchable unit access list associated with the subtask.

To restate, provided herein is a technique for ensuring secured subspaces notwithstanding execution of applications in address register addressing mode. Secured subspaces provide an environment for a server, transaction manager or work manager to provide isolation and protection from multiple concurrent users, transactions or work requests running under separate tasks within a single address space. The server or manager's programs and data may be common to the multiple users and yet isolated and private from other servers or other address spaces allowing for the individual users or task to access the server or manager's functions and yet still have programs and data that are isolated and private to each individual task. If the user or task's application desires to create a multi-tasking environment itself, those additional tasks it creates will share the requesting application's subspace environment, while still being isolated and protected from other user subspace environments. The server or work manager may also use the facilities to provide isolation and protection of certain of its own programs and data from the users or work requests that it is processing.

Additional features and advantages are realized through the techniques of the present invention. Other embodiments and aspects of the invention are described in detail herein and are considered part of the claimed invention.

Brief Description of the Drawings

The above-described objects, advantages and features of the present invention, as well as others, will be more readily understood from the following detailed description of certain preferred embodiments of the invention, when

considered in conjunction with the accompanying drawings in which:

FIG. 1 depicts one example of a block diagram of hardware components of a system to employ a secure subspace facility in accordance with the principles of the present invention;

FIG. 2 depicts one embodiment of a main task and a dispatchable unit access list (DU-AL) structure associated therewith for a conventional transaction manager;

FIGS. 3A & 3B depict a flowchart of one embodiment for creating a secured subspace facility in accordance with the principles of the present invention;

FIG. 4A depicts one embodiment of home/base space storage to be assigned to individual subspaces in accordance with the secured subspace facility embodiment of FIGS. 3A & 3B;

FIG. 4B depicts one embodiment of home space storage and associated subspace assignment in accordance with the secured subspace facility embodiment of FIGS. 3A & 3B;

FIG. 4C depicts one embodiment of a main task, its associated dispatchable unit access list, and home space and subspace assignment in accordance with the secured subspace facility embodiment of FIGS. 3A & 3B;

FIG. 4D depicts one embodiment of the structures of FIG. 4C further depicting creation of a subtask from the main task and an associated DU-AL wherein home space is assigned subspace 1 in accordance with the secured subspace

facility embodiment of FIGS. 3A & 3B and the principles of the present invention;

FIG. 5 depicts one embodiment of a main task, its associated DU-AL and four associated subtasks each having a home space defined in an associated DU-AL as a different subspace in accordance with the secured subspace facility embodiment of FIGS. 3A & 3B and the principles of the present invention;

FIG. 6 graphically depicts one embodiment of secured subspace isolation attained in accordance with the principles of the present invention employing, for example, four subtasks as depicted in FIG. 5; and

FIG. 7 depicts one embodiment of a main task, associated dispatchable unit access list, and multiple subtask generation in accordance with the secured subspace facility embodiment of FIGS. 3A & 3B and the principles of the present invention.

Best Mode for Carrying Out the Invention

In general, this invention defines a software structure which provides a subspace environment for secured subspace isolation and permits access register addressing when secured subspace is active. This is accomplished by defining the content of the dispatchable unit's (DU's) access list's entry (ALET) to contain the subspace's address space number (ASN) second table entry origin (SSASTEO). Subspace tasks can then be created with a secure addressing environment for programs running under those tasks in either primary, secondary or access register addressing mode.

One embodiment of a computing environment incorporating and using the capabilities of the present invention is depicted at FIG. 1. Computing environment 100 is based, for instance, on the Enterprise Systems Architecture (ESA)/390 offered by International Business Machines Corporation, Armonk, New York. ESA/390 is described in an IBM publication entitled Enterprise Systems Architecture/390 Principles of Operation, IBM publication No. SA22-7201-04, June 1997, which is hereby incorporated herein by reference in its entirety.

Computing environment 100 includes, for instance, a main storage 102, one or more central processing units (CPUs) 104 and one or more input/output (I/O) devices 106.

In general, input devices 106 are used to load data and/or programs into main storage 102, and central processing units 104 are used to access the stored programs or data from main storage. Main storage 102 includes one or more address spaces 108, where an address space is a consecutive sequence of integer numbers (or virtual addresses), together with the specific transformation parameters which allow each number to be associated with a byte location in storage. Typically, an entire virtual address space 108 is not resident within main storage. Instead, only that portion associated with a program or data being accessed or used by one or more of the processors is resident within the main storage.

An address space containing a currently dispatched task control block (TCB) or dispatchable unit is referred to herein as a base address space or base space. In a current implementation of the IBM OS/390 operating system, the base space is equivalent to the home address space (home space)

which is described in detail in the above-incorporated IBM Principles of Operation publication. However, in other operating systems, the base space could be distinct from the home space. Also, reference United States Letter Patent No. 5,493,661, by Alpert et al., entitled "Method and System for Providing a Program Call to a Dispatchable Unit's Base Space", the entirety of which is hereby incorporated herein by reference.

FIG. 2 depicts one embodiment of a main task arising under, for example, the above-discussed CICS transaction manager running on an IBM S/390 system. The main task has associated therewith a dispatchable unit - access list (herein referred to as a DU-AL) which identifies the address environment for the main task. In addition to primary and secondary addressing, the access list includes an access list entry (ALE 2) initialized with the home space address space table entry (ASTE) whenever a dispatchable unit is created (e.g., a task, its dispatchable unit control table (DUCT) and the associated DU-AL created through the ATTACH service described in an IBM publication entitled "IBM OS/390 MVS Assembler Services Reference", publication no. GC 28-1910-09, the entirety of which is hereby incorporated herein by reference). ALE 2 initialized with the home space ASTE enables any program running in access register mode to access the entire home space, within the boundaries of the key protection facilities. Typically, the home space and the base space comprise the same space. Addressing ranges in the base space can be reserved and separate ranges allocated exclusively to individual subspaces (defined in the above-incorporated S/390 Principles of Operation). This limits the addressing scope of programs running with subspace active to the ranges reserved for their current subspace. However, this isolation does not apply when the

program runs in access register addressing mode. An ALET 2 in an access register (AR) qualified address provides the program addressing access to the home space, hence the base space and to all areas that the subspace should not be privileged to access. As noted above, the solution to this problem has been to restrict the transaction manager such that when in secured base space mode, access register addressing mode is unavailable.

In contrast, the solution presented herein achieves secured subspace isolation and allows access register addressing by attaching a subtask that will restrict application addressing. Specifically, the attaching includes defining a subspace address environment as home space within the dispatchable unit access list (DU-AL) associated with the subtask. By initializing the ALE 2 value to the subspace ASTE origin instead of the home space ASTE when the subtask and its DU-AL are created and initialized, the subtask is unable to access home space of the main task and therefore access register addressing can be employed. The addressability of any program running with subspace active will then be limited to its subspace addressing environment whether the program is running in primary or access register addressing mode. In one embodiment, the IBM S/390 ATTACH service is modified (as described below) to support an option to create tasks with a subspace addressing environment. The attaching task's current subspace environment then will be the subspace addressing environment used to initialize the attached task's DU-AL ALE 2 value to the subspace's ASTE origin.

In an OS/390 implementation, this structure is possible because the content of ALE 2 of the DU-AL is an S/390 software convention. This does not effect control register

13 (i.e., a hardware control register defined in the IBM S/390 Principles of Operations) which contains the home space segment table descriptor (STD). The CR 13 home space STD is architecturally defined and implemented by hardware
5 to be used for address and instruction accesses whenever running in home space mode. Since programs must be authorized to SAC (i.e., a hardware instruction "Set Address Space Control" defined in the IBM S/390 Principles of Operations) to home-space mode, the secured subspace concept
10 of isolation for problem state programs is not effected by keeping the current architecture definition of CR 13.

Note that the present invention is related in one aspect to the addressing capabilities within a subspace active environment. Architecture and hardware for a branch
15 in subspace group is discussed in the above-incorporated U.S. Letters Patent No. 5,361,356.

FIGS. 3A & 3B depict one embodiment for creating a secure subspace environment in accordance with the principles of the present invention. A task management
20 function running under a main task can create a secured subspace environment for each transaction to be run in a transaction isolation environment wherein the transactions are unable to access each others' assigned memory. Under the main task, the transaction manager creates n subspaces
25 by first obtaining storage in the home space to be assigned to individual subspaces 300. Note that as one embodiment, the functions depicted in FIGS. 3A & 3B and described herein are available with IBM's System 390 operating system, i.e., unless otherwise indicated. Also note that there is one
30 home space/base space per CICS region that is started in the S/390 system. Next, the storage range eligible for subspace assignment is identified, for example, using the IBM OS/390

"IARSUBSP" service 310. All other storage will be available to the base space and its subspaces as storage is obtained. FIG. 4A depicts one embodiment of a home/base space showing four different areas of storage obtained and identified.

5 Next, a subspace is created, for example, using the "IARSUBSP" create service of the OS/390 system 320. A space token (STOKEN) is received that represents the subspace. Note that the identified range is not made available to the subspace. FIG. 4B depicts one embodiment of a home space
10 having four ranges A, B, C & D and an associated subspace showing that the corresponding ranges are not yet addressable in the subspace.

 A subspace is thereafter added to the main task's DU-
AL, for example, using the OS/390 "ALESERV" (access list
15 entry) ADD service 330. An ALET is received back that represents the subspace, for example, ALET 3 as shown in FIG. 4C, wherein the subspace is now labeled subspace 1.

 A range of storage that transactions running in the subspace can access is assigned, e.g., using the OS/390
20 "IARSUBSP" assign service 340. For example, the service allocates and assigns storage A and makes that valid in subspace 1 as shown in FIG. 4C, wherein the crosshatch signifies not valid.

 A branch specifying the access list entry (ALE) to the
25 desired function is next performed which makes the subspace the active addressing environment. In one embodiment, this comprises a BSG hardware instruction 350. Note that after the BSG instruction, the subspace (for example, subspace 1 of FIG. 4C) becomes the active primary and secondary address
30 space for the task. All instructions and data accesses from

primary or secondary addressing will come from and be limited to the subspace's addressing range. However, an application at this point in access register addressing mode can still address the home space via ALET 2. This issue is
5 addressed by the present invention.

As noted above, the solution presented herein to attach a subtask that will restrict application addressing 360. A subtask can be attached for each transaction to be run using secured subspaces. Each attached subtask has a subspace
10 address environment as home space within the dispatchable unit access list associated with that subtask. The result is shared subspaces between the subtask and the main task, but isolated subspaces between independently attached subtasks. The ability to share subspaces is provided, in
15 one example, through the IBM OS/390 Attach service described in the above-incorporated IBM OS/390 MVS Assembler Services Reference publication. However, a new parameter "ADDRENV" is defined as described herein for the ATTACHX macro. The ATTACHX macro is a means for programs to request the IBM
20 OS/390 Attach Service. This new parameter permits a task that has created a subspace to attach a subtask and limit the addressability of the subtask to the addressing environment defined by the subspace. Subtasks attached with this subspace limited addressability can be designated as
25 "subspace tasks". Subspace tasks can only attach tasks that are also subspace tasks. This new ATTACHX parameter, supports two options, namely, ADDRENV=SAME and ADDRENV=SUBSP. ADDRENV=SAME specifies that the attached task will be attached with the same primary mode addressing
30 environment of the attaching task. The primary mode addressing environment is exclusive of the addressing environment that can be created through the IBM OS/390 ALCOPY service processing for the DU-AL addressing

environment. The attached task's home ALET (ALET 2) ASTE designation for an ADDRENV=SAME request will be the same ASTE designation as the home ALET of the attaching task.

5 ADDRENV=SUBSP specifies that the attached task will be attached with the subspace addressing environment that was active when the attach was issued. The task issuing the ATTACHX request must own the subspace and establish the subspace as the current active subspace before issuing the ATTACHX. The new task will be attached with subspace active
10 and have an addressing environment limited to the addresses accessible to that subspace. The ATTACH service will designate the task as a subspace task.

15 These subspace tasks are limited to attaching only subspace tasks with the SAME or a new SUBSP addressing environment. A subspace task cannot attach a base/space task, e.g., a task that is not a subspace task.

20 When a task is attached with ADDRENV=SUBSP, the attached task will receive control with the SUBSPACE as the active subspace. The home ALET of the task will be set to the SUBSPACE'S ASTE. An AR-qualified address specifying the home ALET will designate the subspace and its addressing environment. A task running with subspace-active is either the owner of the subspace or is an attached subspace task created by the owner of the subspace. A subspace task could
25 also attach another subspace task with the subspace being shared with that task. However, the owning task of the subspace is still the parent or guardian task of any subspace task using that subspace. Therefore, the owning task cannot terminate and delete the subspace until the
30 subspace tasks using the subspace have terminated.

This structure guarantees that a subspace owning task will not terminate and attempt to delete the subspace if there are any subtasks actively using the subspace. This simplifies the process and serialization for managing the
5 deletion of subspaces.

By way of example, in the IBM CICS transaction server open transaction environment (OTE), an open CICS application has the option to be run under an OTE task. This task will be created as a subspace task by the CICS main or quasi/re-
10 entrant (Q/R) task. The Q/R task will create many OTE tasks and each one will be created as a subspace task with its own subspace to provide transaction isolation between the multiple transactions. The Q/R task will own the subspaces that define the addressing environments for the open
15 transactions. A transaction will run under an OTE task with the subspace protection. However, if the transaction requests a CICS function that has not been rewritten as a re-entrant function, CICS will move the transaction to the Q/R task and run the function under the Q/R task. When this
20 transaction runs under the Q/R, its subspace environment will be made active by CICS identifying the subspace on its DU-AL (the Q/R task's dispatchable unit's access list) that is associated with the transaction and then branching (BSG) to that subspace. Subspace sharing allows this to be done.
25 Without subspace sharing, the subspace's designation would have to be moved from the OTE task's access list to the Q/R task's access list and vise/versa.

Returning to FIG. 3B, by using the above-described attach function, a subtask is created that will restrict
30 application addressing in primary, secondary and access register mode to the subspace. By way of example, this structure can be attained using the IBM OS/390 ATTACHX

service discussed above wherein the home space of the dispatchable unit access list associated with the subtask is assigned the subspace address environment (reference FIG. 4D). Note that when the application receives control

5 running under the subtask, the application will be limited such that it cannot access other subspaces/transaction areas (e.g., B, C & D in FIG. 4D) when employing primary, secondary or access register mode addressing and when subspace is active. The utilization of the subspace as the

10 subtask's home space (ALET 2) permits base control programs to continue to access base control program (BCP) structures using ALET 2, but restricts addressing storage not assigned to the subspace. Hence, the application running under the subtask will not be able to access other transactions/

15 application addressable areas.

The processing of FIGs. 3A & 3B, and in particular STEPS 320-360, is repeated for each subspace environment required to provide transaction isolation, 370.

FIG. 5 depicts one example of the results from

20 repeating STEPS 320-360 for four different subtasks, and shows their secured subspace environments such that transaction/applications running under the individual subtasks are restricted from accessing the assigned storage of the transactions/applications running under different

25 subtasks. For example, as shown in FIG. 6, applications under subtask1 cannot address assigned storage areas B, C & D. Applications under subtask2 cannot address assigned storage areas A, C & D. Applications under subtask3 cannot address assigned storage areas A, B & D and applications

30 under subtask4 cannot address assigned storage areas A, B & C. Again, each defined subtask has an associated

dispatchable unit access list with the corresponding subspace defined as home space in ALET 2 as shown in FIG. 5.

FIG. 7 depicts one embodiment of the main task and its associated access list of FIG. 1, which again employs secured subspaces in accordance with the principles of the present invention. In this embodiment, the main task attaches a first subtask (task A1) and a second subtask (task B) directly, for example, using the above-described modified ATTACHX service of the IBM OS/390 system. The address environment in these attaches is defined as ADDRENV=SUBSP. In both the access list for task A1 and the access list for task B, the home space is defined as a corresponding subspace (SS1, SS2 respectively). A transaction running under task A1 can itself create a subtask, for example, task A2. However, task A2 is defined to have the same address environment as task A1, i.e., subspacel. This feature is advantageous in that applications can create a multi-tasking environment themselves. All subtasks created in this environment are limited to sharing the subspace of the attaching task.

The present invention can be included in an article of manufacture (e.g., one or more computer program products) having, for instance, computer usable media. The media has embodied therein, for instance, computer readable program code means for providing and facilitating the capabilities of the present invention. The article of manufacture can be included as a part of a computer system or sold separately.

Additionally, at least one program storage device readable by a machine, tangibly embodying at least one program of instructions executable by the machine to perform the capabilities of the present invention can be provided.

The flow diagrams depicted herein are just examples. There may be many variations to these diagrams or the steps (or operations) described therein without departing from the spirit of the invention. For instance, the steps may be performed in a differing order, or steps may be added,
5 deleted or modified. All of these variations are considered a part of the claimed invention.

Although preferred embodiments have been depicted and described in detail herein, it will be apparent to those
10 skilled in the relevant art that various modifications, additions, substitutions and the like can be made without departing from the spirit of the invention and these are therefore considered to be within the scope of the invention as defined in the following claims.

Claims

1 1. A method for producing a secure subspace for a
2 transaction, said method comprising:

3 from an operating system task, attaching a subtask
4 that will restrict application addressing; and

5 wherein said attaching includes defining a
6 subspace address environment as home space within a
7 dispatchable unit access list (DU-AL) associated with
8 said subtask.

1 2. The method of claim 1, wherein said subtask
2 comprises a first subtask, said subspace comprises a first
3 subspace and a first application runs under said first
4 subtask, and wherein said method further comprises repeating
5 said attaching to define a second subtask having a second
6 subspace address environment as home space within a DU-AL
7 associated with said second subtask, wherein a second
8 application runs under said second subtask.

1 3. The method of claim 2, wherein said first subspace
2 is isolated from said second application and said second
3 subspace is isolated from said first application
4 notwithstanding execution of said first application or said
5 second application in address register addressing mode.

1 4. The method of claim 2, wherein said operating
2 system task and said first subtask share said first
3 subspace, and said operating system task and said second
4 subtask share said second subspace.

1 5. The method of claim 2, further comprising
2 repeating said subtask attaching for n additional subtasks,
3 each subtask of said n additional subtasks having a
4 different subspace address environment as home space within
5 its associated DU-AL, wherein each subspace of said first,
6 second and n additional subtasks is isolated from an
7 application running under any other subtask of said first,
8 second and n additional subtasks.

1 6. The method of claim 5, wherein each subspace
2 address environment of said first, second and n additional
3 subtasks comprises a different subspace of an address
4 environment of said operating system task.

1 7. The method of claim 1, further comprising prior to
2 said attaching:

3 creating said subspace;

4 adding said subspace to a DU-AL associated with
5 said operating system task;

6 assigning a range of storage that an application
7 running in the subspace can access; and

8 performing a branch in subspace group (BSG) to
9 make the subspace the active addressing environment.

1 8. The method of claim 7, wherein said performing the
2 BSG comprises employing a BSG instruction to specify an
3 access list entry (ALET) in the DU-AL associated with said
4 operating system task.

1 9. The method of claim 1, wherein said subtask
2 comprises a first subtask and a first application runs under
3 said first subtask and wherein said method further comprises
4 creating a second subtask from said first subtask, said
5 creating comprising from said first subtask, attaching said
6 second subtask thereto, said second subtask also having said
7 subspace address environment as home space within a DU-AL
8 associated therewith, wherein said subspace is shared by
9 said operating system task, said first subtask and said
10 second subtask.

1 10. The method of claim 9, wherein said subspace
2 comprises a first subspace, and said method further
3 comprises repeating said attaching from said operating
4 system task to define a third subtask having a second
5 subspace address environment as home space within a DU-AL
6 associated with said third subtask, wherein a second
7 application runs under said third subtask, and wherein said
8 first application and said second application are unable to
9 access each other's address environment notwithstanding
10 execution thereof in address register addressing mode.

1 11. At least one program storage device readable by a
2 machine, tangibly embodying at least one program of
3 instructions executable by the machine to perform a method
4 for producing a secure subspace for a transaction, said
5 method comprising:

6 from an operating system task, attaching a subtask
7 that will restrict application addressing; and

8 wherein said attaching includes defining a
9 subspace address environment as home space within a
10 dispatchable unit access list (DU-AL) associated with
11 said subtask.

1 12. The at least one program storage device of claim
2 11, wherein said subtask comprises a first subtask, said
3 subspace comprises a first subspace and a first application
4 runs under said first subtask, and wherein said method
5 further comprises repeating said attaching to define a
6 second subtask having a second subspace address environment
7 as home space within a DU-AL associated with said second
8 subtask, wherein a second application runs under said second
9 subtask.

1 13. The at least one program storage device of claim
2 12, wherein said first subspace is isolated from said second
3 application and said second subspace is isolated from said
4 first application notwithstanding execution of said first
5 application or said second application in address register
6 addressing mode.

1 14. The at least one program storage device of claim
2 12, wherein said operating system task and said first
3 subtask share said first subspace, and said operating system
4 task and said second subtask share said second subspace.

1 15. The at least one program storage device of claim
2 12, further comprising repeating said subtask attaching for
3 n additional subtasks, each subtask of said n additional
4 subtasks having a different subspace address environment as
5 home space within its associated DU-AL, wherein each
6 subspace of said first, second and n additional subtasks is
7 isolated from an application running under any other subtask
8 of said first, second and n additional subtasks.

1 16. The at least one program storage device of claim
2 15, wherein each subspace address environment of said first,
3 second and n additional subtasks comprises a different
4 subspace of an address environment of said operating system
5 task.

1 17. The at least one program storage device of claim
2 11, further comprising prior to said attaching:

3 creating said subspace;

4 adding said subspace to a DU-AL associated with
5 said operating system task;

6 assigning a range of storage that an application
7 running in the subspace can access; and

8 performing a branch in subspace group (BSG) to
9 make the subspace the active addressing environment.

1 18. The at least one program storage device of claim
2 17, wherein said performing the BSG comprises employing a
3 BSG instruction to specify an access list entry (ALET) in
4 the DU-AL associated with said operating system task.

1 19. The at least one program storage device of claim
2 11, wherein said subtask comprises a first subtask and a
3 first application runs under said first subtask and wherein
4 said method further comprises creating a second subtask from
5 said first subtask, said creating comprising from said first
6 subtask, attaching said second subtask thereto, said second
7 subtask also having said subspace address environment as
8 home space within a DU-AL associated therewith, wherein said
9 subspace is shared by said operating system task, said first
10 subtask and said second subtask.

1 20. The at least one program storage device of claim
2 19, wherein said subspace comprises a first subspace, and
3 said method further comprises repeating said attaching from
4 said operating system task to define a third subtask having
5 a second subspace address environment as home space within a
6 DU-AL associated with said third subtask, wherein a second
7 application runs under said third subtask, and wherein said
8 first application and said second application are unable to
9 access each other's address environment notwithstanding
10 execution thereof in address register addressing mode.

1 21. A system for producing a secure subspace for a
2 transaction, said system comprising:

3 means for attaching, from an operating system
4 task, a subtask that will restrict application
5 addressing; and

6 wherein said means for attaching includes means
7 for defining a subspace address environment as home
8 space within a dispatchable unit access list (DU-AL)
9 associated with said subtask.

1 22. The system of claim 21, wherein said subtask
2 comprises a first subtask, said subspace comprises a first
3 subspace and a first application runs under said first
4 subtask, and wherein said system further comprises means for
5 repeating said attaching to define a second subtask having a
6 second subspace address environment as home space within a
7 DU-AL associated with said second subtask, wherein a second
8 application runs under said second subtask.

1 23. The system of claim 22, wherein said first
2 subspace is isolated from said second application and said
3 second subspace is isolated from said first application
4 notwithstanding execution of said first application or said
5 second application in address register addressing mode.

1 24. The system of claim 22, wherein said operating
2 system task and said first subtask share said first
3 subspace, and said operating system task and said second
4 subtask share said second subspace.

1 25. The system of claim 22, further comprising means
2 for repeating said subtask attaching for n additional
3 subtasks, each subtask of said n additional subtasks having
4 a different subspace address environment as home space
5 within its associated DU-AL, wherein each subspace of said
6 first, second and n additional subtasks is isolated from an
7 application running under any other subtask of said first,
8 second and n additional subtasks.

1 26. The system of claim 25, wherein each subspace
2 address environment of said first, second and n additional
3 subtasks comprises a different subspace of an address
4 environment of said operating system task.

1 27. The system of claim 21, further comprising prior
2 to said means for attaching:

3 means for creating said subspace;

4 means for adding said subspace to a DU-AL
5 associated with said operating system task;

6 means for assigning a range of storage that an
7 application running in the subspace can access; and

8 means for performing a branch in subspace group
9 (BSG) to make the subspace the active addressing
10 environment.

1 28. The system of claim 27, wherein said means for
2 performing the BSG comprises means for employing a BSG
3 instruction to specify an access list entry (ALET) in the
4 DU-AL associated with said operating system task.

1 29. The system of claim 21, wherein said subtask
2 comprises a first subtask and a first application runs under
3 said first subtask and wherein said system further comprises
4 means for creating a second subtask from said first subtask,
5 said means for creating comprising means for attaching said
6 second subtask to said first subtask, said second subtask
7 also having said subspace address environment as home space
8 within a DU-AL associated therewith, wherein said subspace
9 is shared by said operating system task, said first subtask
10 and said second subtask.

1 30. The system of claim 29, wherein said subspace
2 comprises a first subspace, and said system further
3 comprises means for repeating said means for attaching from
4 said operating system task to define a third subtask having
5 a second subspace address environment as home space within a
6 DU-AL associated with said third subtask, wherein a second
7 application runs under said third subtask, and wherein said
8 first application and said second application are unable to
9 access each other's address environment notwithstanding
10 execution thereof in address register addressing mode.

1 31. A system for producing a secure subspace for a
2 transaction, said system comprising:

3 an operating system transaction manager adapted to
4 attach a subtask to an operating system task, wherein
5 said subtask restricts application addressing; and

6 wherein said attach includes said operating system
7 transaction manager being adapted to define a subspace
8 address environment as home space within a dispatchable
9 unit access list (DU-AL) associated with said subtask.

* * * * *

**STORAGE ISOLATION EMPLOYING
SECURED SUBSPACE FACILITY**

Abstract of the Disclosure

5 A secured subspace facility is provided for ensuring
isolated storage for transactions running under an operating
system main task. Isolation is achieved by attaching, from
an operating system task, subtasks that will restrict
application addressing. The attaching includes defining a
subspace address environment as home space within a
10 dispatchable unit access list (DU-AL) associated with each
attached subtask. Multiple subtasks can be attached with
each subtask running applications in an isolated address
subspace, notwithstanding execution of the applications in
address register addressing mode.

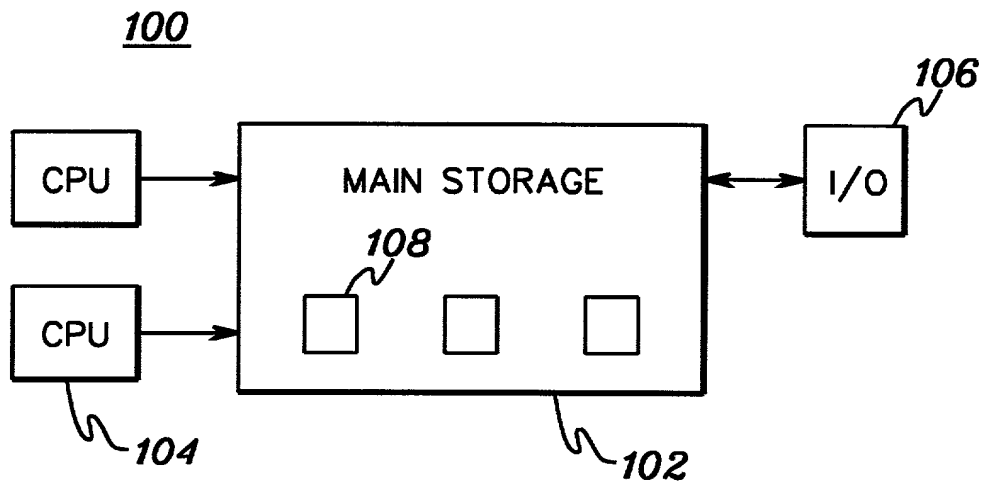


fig. 1

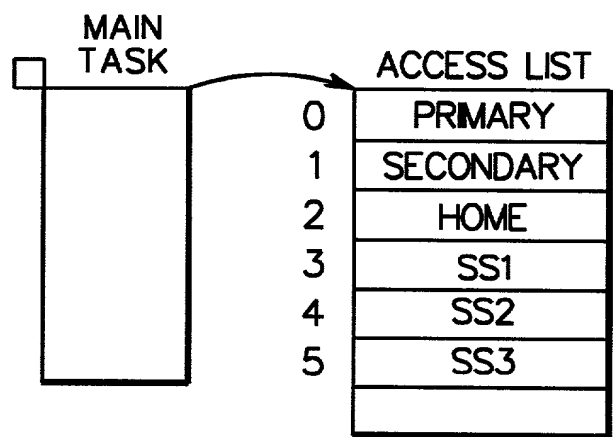
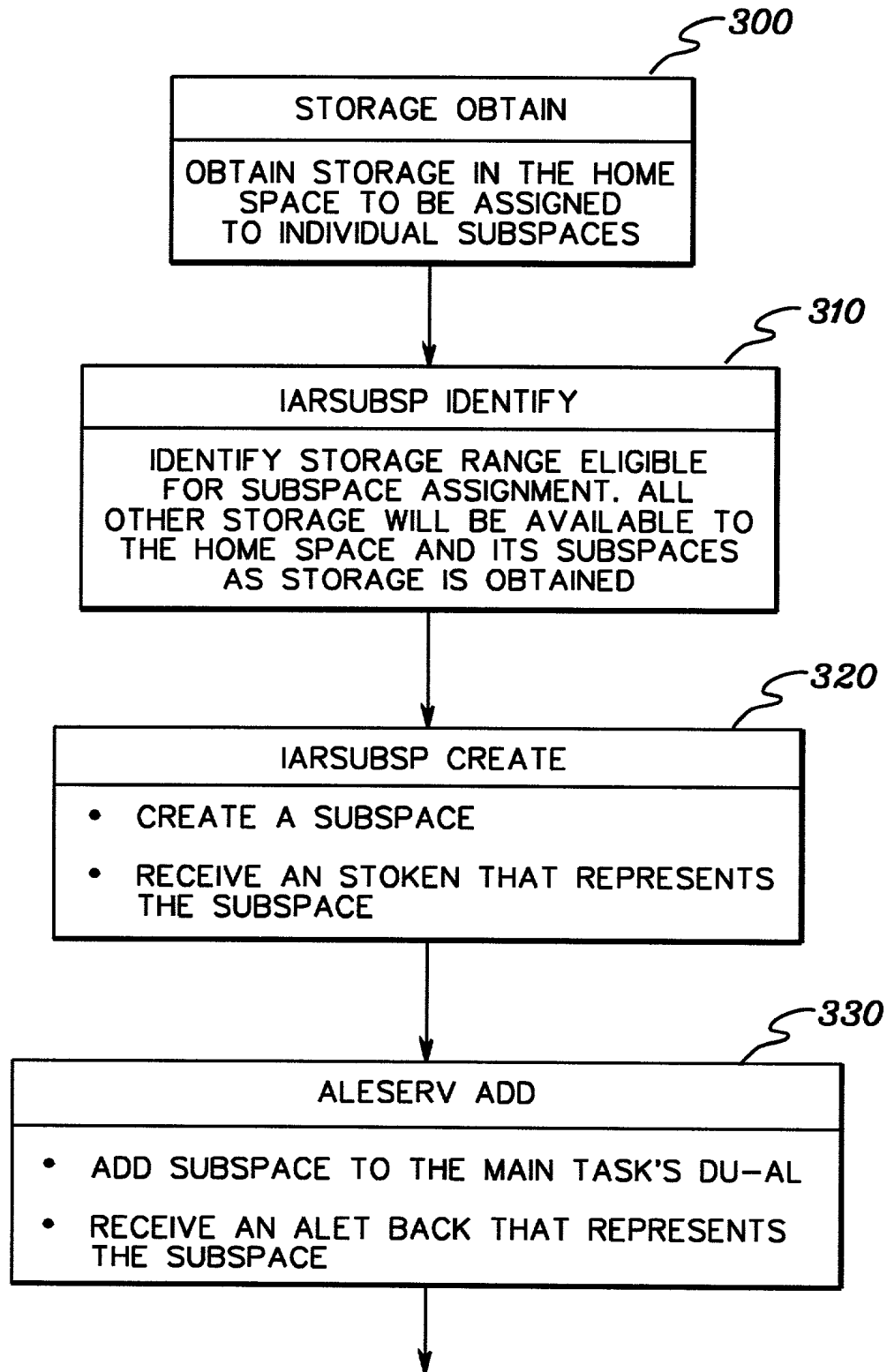


fig. 2



TO FIG. 3B

fig. 3A

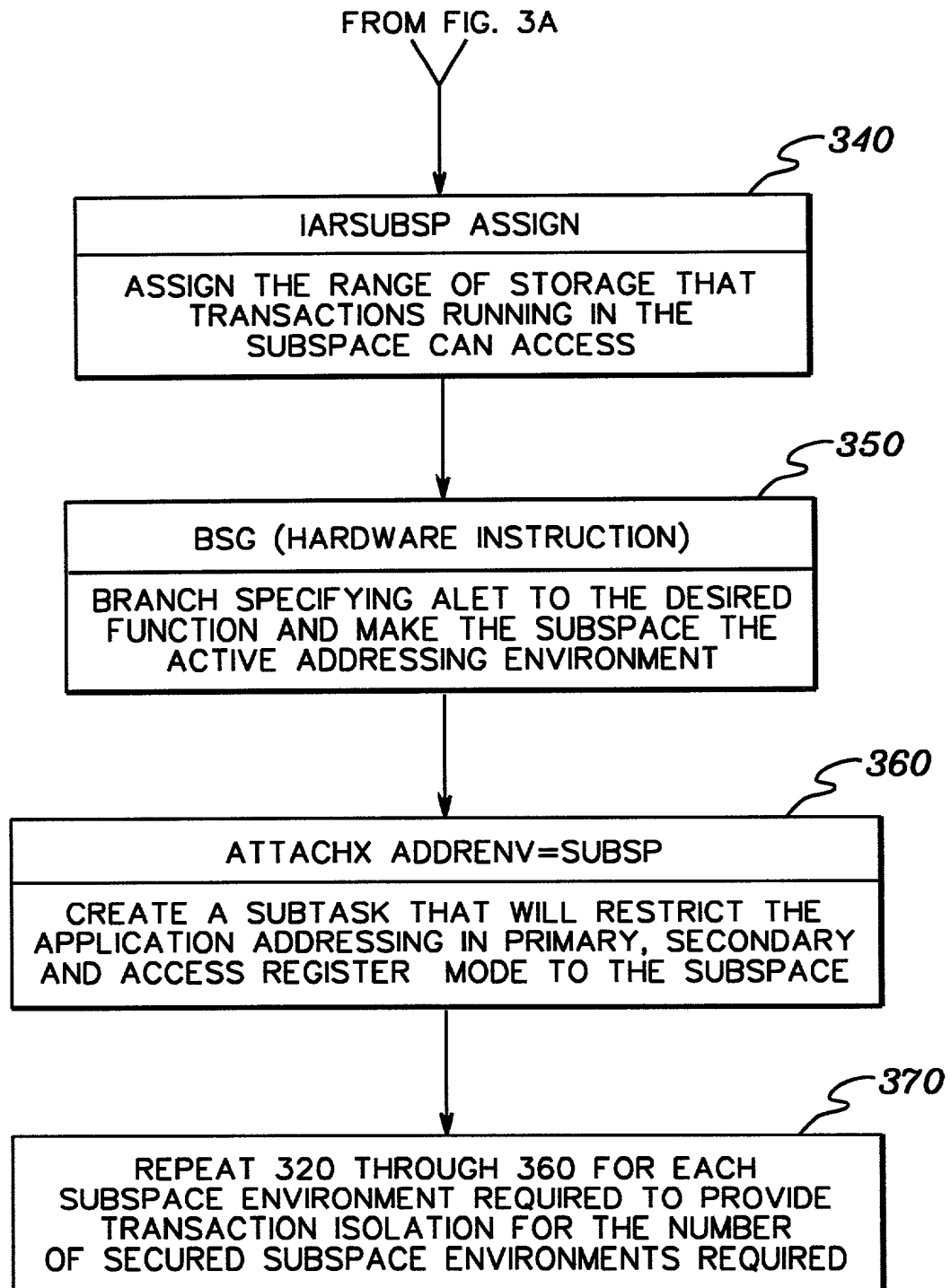


fig. 3B

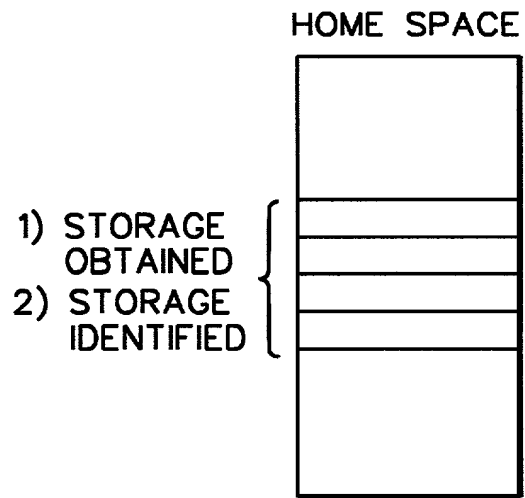


fig. 4A

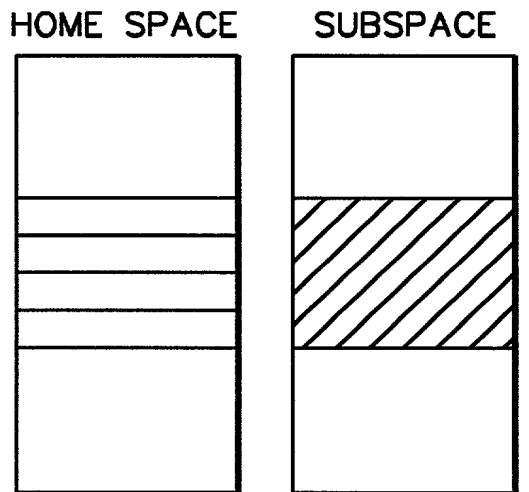


fig. 4B

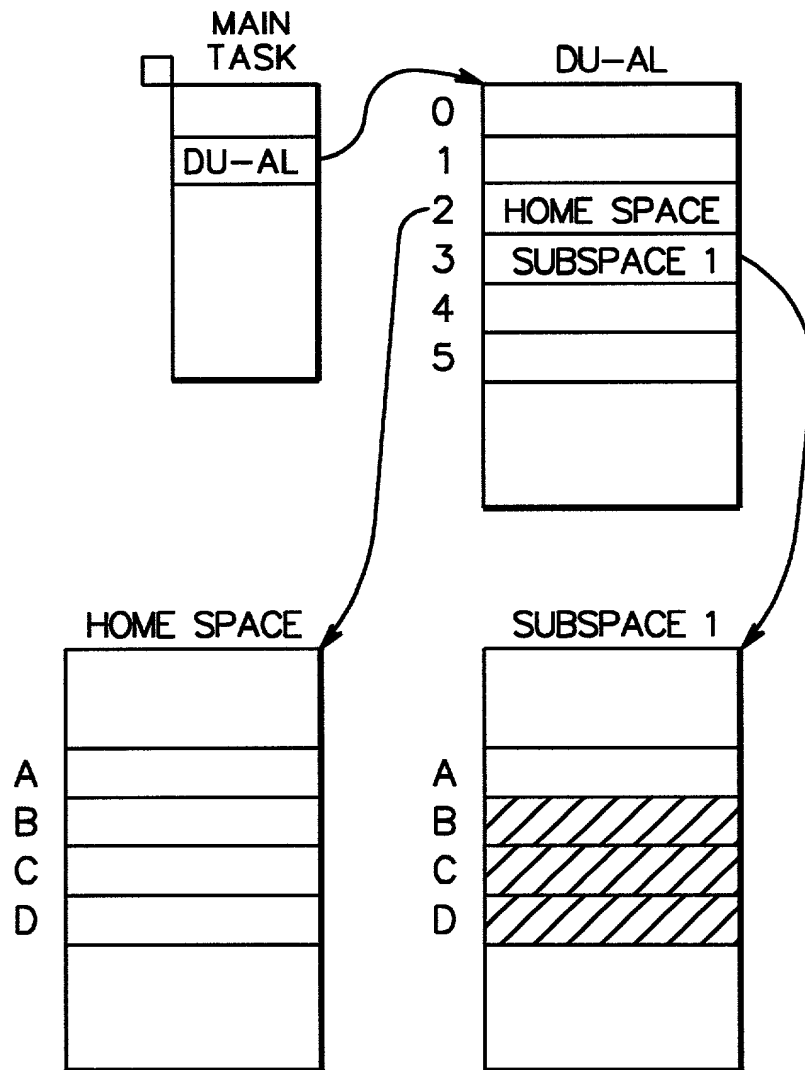


fig. 4C

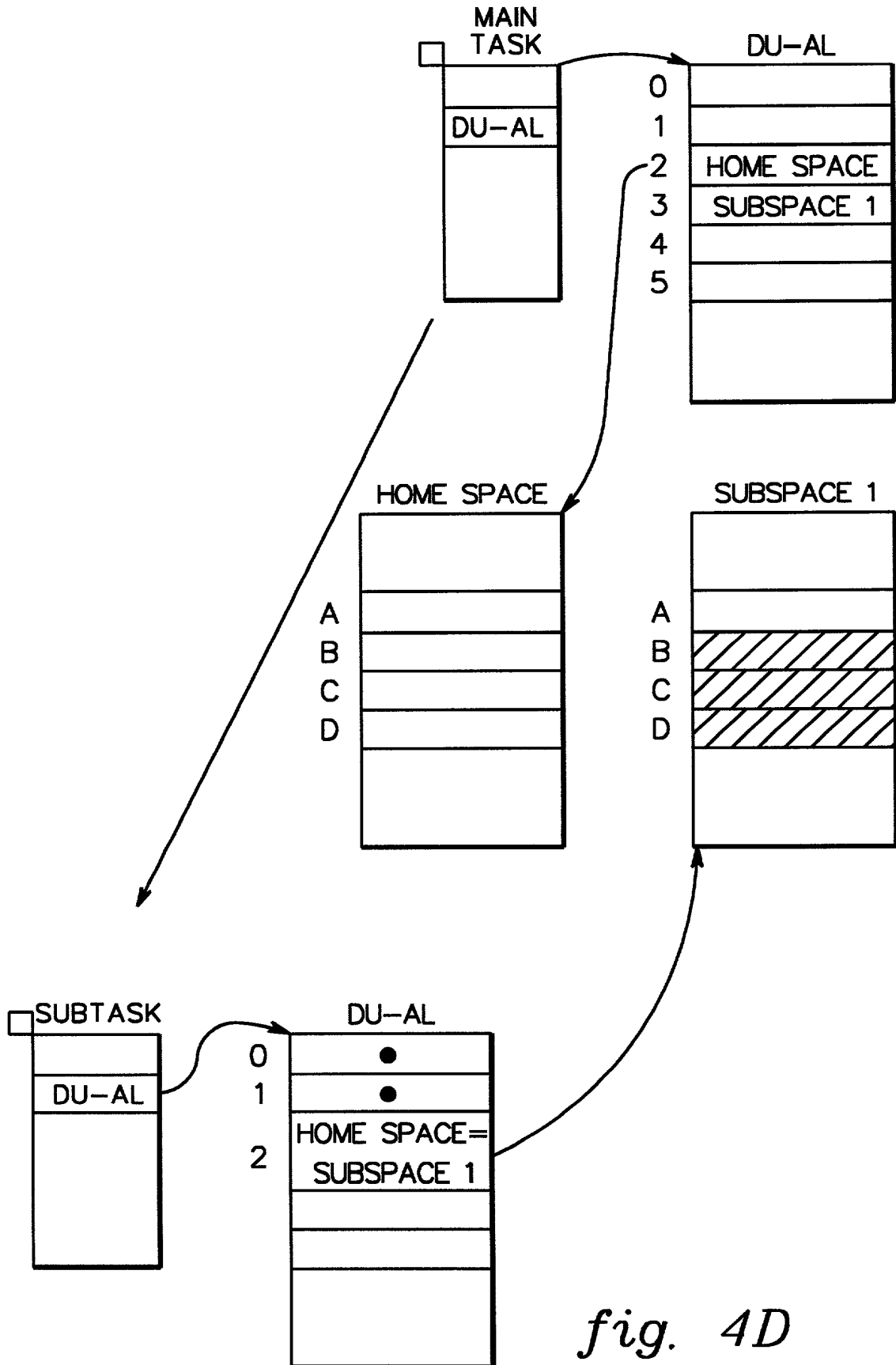


fig. 4D

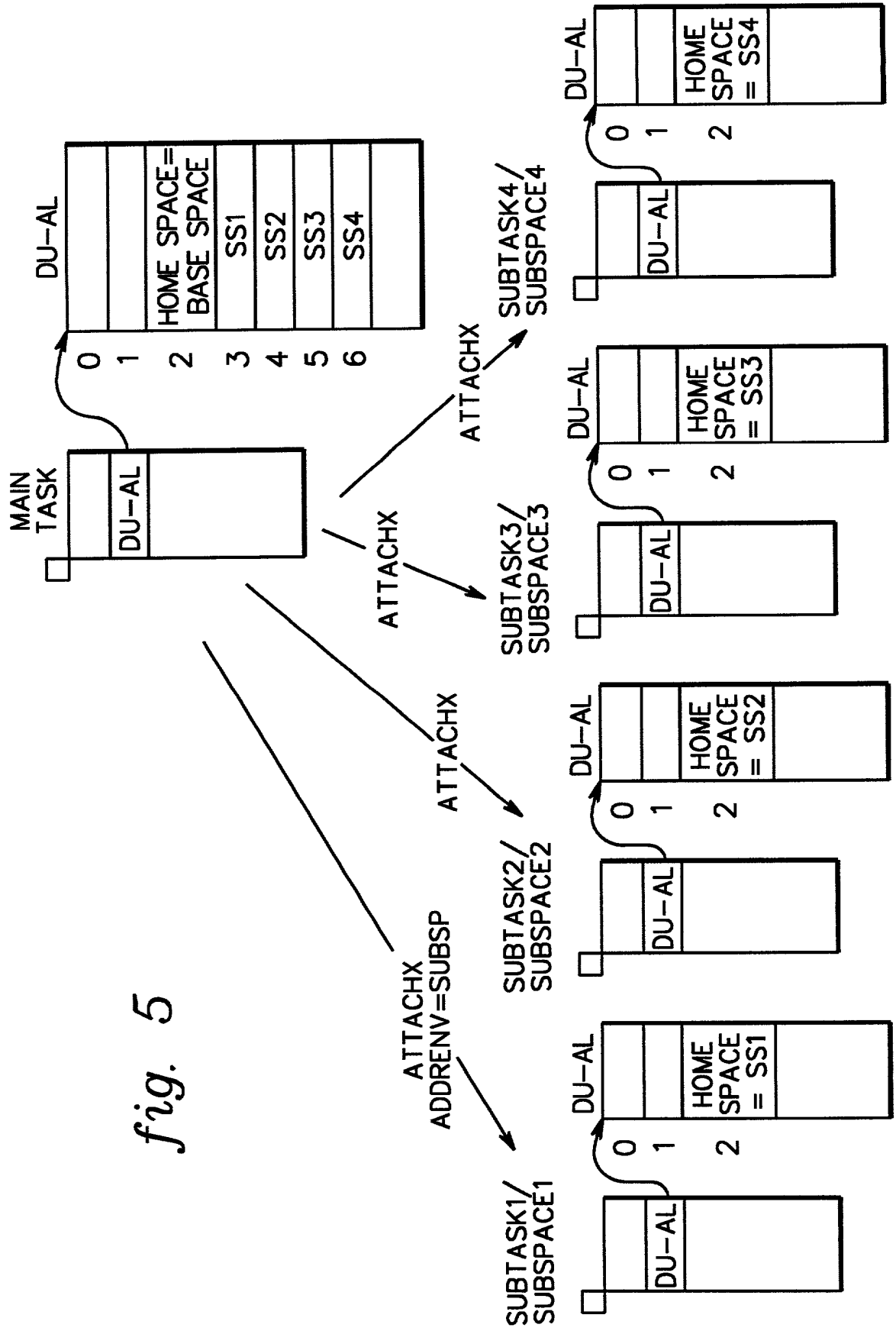


fig. 5

POU9-2000-0030-US1

	HOME SPACE	SS1	SS2	SS3	SS4
A					
B					
C					
D					

fig. 6

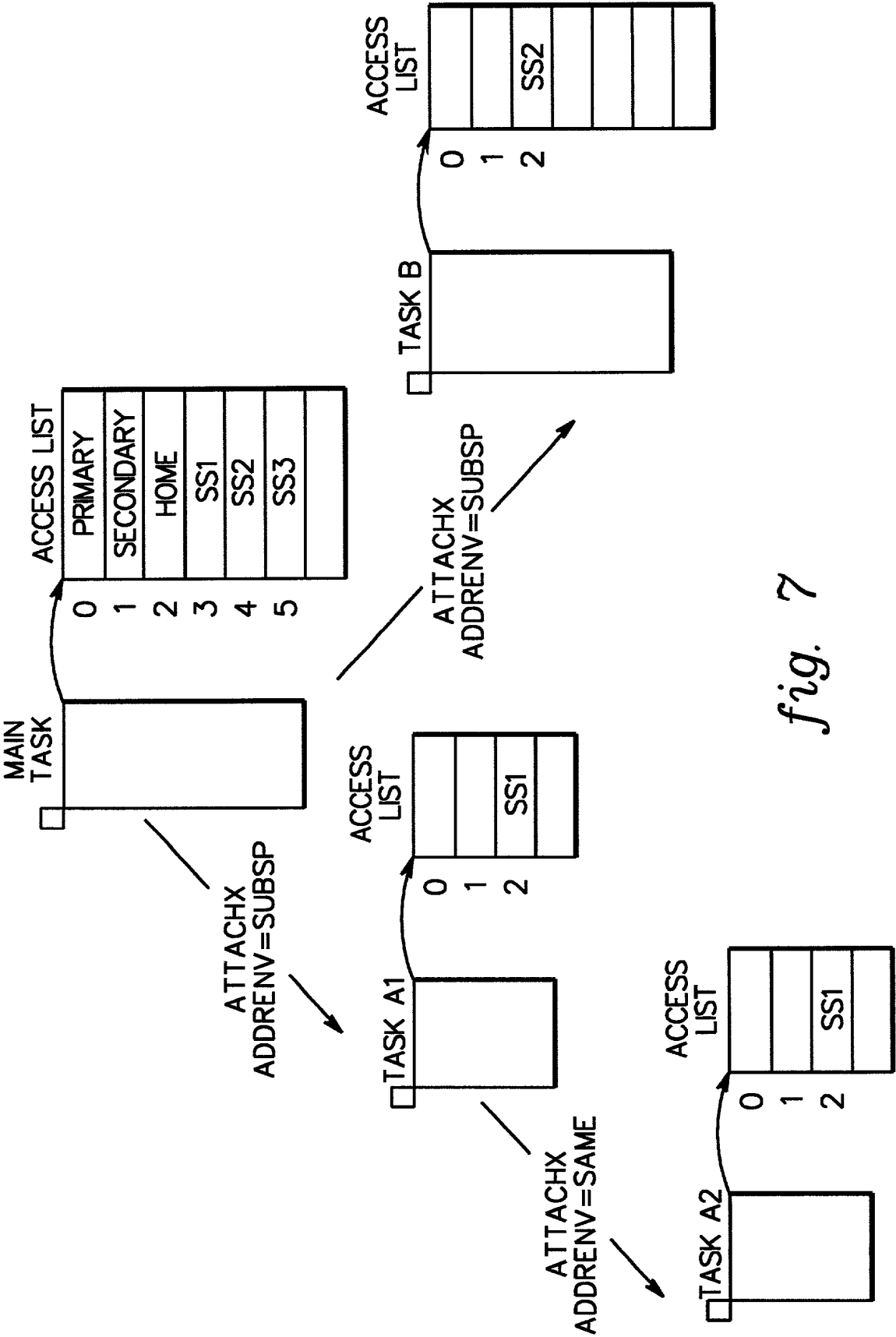


fig. 7

Docket No.

POU9-2000-0030-US1

Declaration and Power of Attorney For Patent Application

English Language Declaration

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name,

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

STORAGE ISOLATION EMPLOYING SECURED SUBSPACE FACILITY

the specification of which

(check one)

☒ is attached hereto.

☐ was filed on _____ as United States Application No. or PCT International Application Number _____ and was amended on _____ (if applicable)

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose to the United States Patent and Trademark Office all information known to me to be material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, Section 119(a)-(d) or Section 365(b) of any foreign application(s) for patent or inventor's certificate, or Section 365(a) of any PCT International application which designated at least one country other than the United States, listed below and have also identified below, by checking the box, any foreign application for patent or inventor's certificate or PCT International application having a filing date before that of the application on which priority is claimed.

Prior Foreign Application(s)

Priority Not Claimed

(Number) (Country) (Day/Month/Year Filed)

☐

(Number) (Country) (Day/Month/Year Filed)

☐

(Number) (Country) (Day/Month/Year Filed)

☐

I hereby claim the benefit under 35 U.S.C. Section 119(e) of any United States provisional application(s) listed below:

(Application Serial No.)

(Filing Date)

(Application Serial No.)

(Filing Date)

(Application Serial No.)

(Filing Date)

I hereby claim the benefit under 35 U. S. C. Section 120 of any United States application(s), or Section 365(c) of any PCT International application designating the United States, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT International application in the manner provided by the first paragraph of 35 U.S.C. Section 112, I acknowledge the duty to disclose to the United States Patent and Trademark Office all information known to me to be material to patentability as defined in Title 37, C. F. R., Section 1.56 which became available between the filing date of the prior application and the national or PCT International filing date of this application:

(Application Serial No.)

(Filing Date)

(Status)
(patented, pending, abandoned)

(Application Serial No.)

(Filing Date)

(Status)
(patented, pending, abandoned)

(Application Serial No.)

(Filing Date)

(Status)
(patented, pending, abandoned)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith. *(list name and registration number)*

William B. Porter, Reg. No. 33,135

Floyd A. Gonzalez, Reg. No. 26,732

Lynn L. Augspurger, Reg. No. 24,227

William A. Kinnaman, Jr., Reg. No. 27,650

Lily Neff, Reg. No. 38,254

Marc A. Ehrlich, Reg. No. 39,966

Lawrence D. Cutter, Reg. No. 28,501

Christopher A. Hughes, Reg. No. 26,914

Edward A. Pennington, Reg. No. 32,588

John E. Hoel, Reg. No. 26,279

Joseph C. Redmond, Reg. No. 18,753

Jeff Rothenberg, Reg. No. 26,429

Kevin P. Radigan, Reg. No. 31,789

Blanche E. Schiller, Reg. 35,670

Send Correspondence to: **Kevin P. Radigan, Esq.**
HESLIN & ROTHENBERG, P.C.
5 Columbia Circle
Albany, NY 12203

Direct Telephone Calls to: *(name and telephone number)*
Kevin P. Radigan, Esq. (518) 452-5600

Full name of sole or first inventor CARL E. CLARK	
Sole or first inventor's signature	Date
Residence 46 Bart Drive, Poughkeepsie, NY 12603	
Citizenship United States of America	
Post Office Address 46 Bart Drive, Poughkeepsie, NY 12603	

Full name of second inventor, if any STEVEN J. GREENSPAN	
Second inventor's signature	Date
Residence 25 Fallkill Road, Hyde Park, NY 12538	
Citizenship United States of America	
Post Office Address 25 Fallkill Road, Hyde Park, NY 12538	